階層的複数ラベル文書分類におけるラベル間依存の利用

村脇　有吾†

階層的複数ラベル文書分類においては，あらかじめ定義されたラベル階層の利用が中心的な課題となる．本稿では，複数の出力ラベル間の依存関係という，従来研究が用いてこなかった手がかりを利用する手法を提案する．これを実現するために，まずはこのタスクを構造推定問題として定式化し，複数のラベルを同時に出力する大域モデルと，動的計画法による厳密解の探索手法を提案する．次に，ラベル間依存を表現する枝分かれ特徴量を導入する．実験では，ラベル間依存の特徴量の導入により，精度の向上とともに，モデルの大きさの削減が確認された．

キーワード：文書分類，構造推定問題，木，動的計画法，オンライン学習

Exploiting Inter-label Dependencies in Hierarchical Multi-Label Document Classification

Yugo Murawaki†

The main challenge in hierarchical multi-label document classification is the means by which hierarchically organized labels are leveraged. In this paper, we propose to exploit dependencies among multiple labels to be output, which has not been considered in previous studies. To accomplish this, we first formalize the task as a structured prediction problem and propose (1) a global model that jointly outputs multiple labels and (2) a decoding algorithm that finds an exact solution with dynamic programming. We then introduce features that capture inter-label dependencies. Experiments show that these features improve performance while reducing the model size.

Key Words: *document classification, structured prediction problem, tree, dynamic programming, online learning*

1　はじめに

電子化されたテキストが利用可能になるとともに，階層的文書分類の自動化が試みられてきた．階層的分類の対象となる文書集合の例としては，特許[1]，医療オントロジー[2]，Yahoo!や Open Directory Project[3]のようなウェブディレクトリが挙げられる．文書に付与すべきラベルは，タ

† 九州大学大学院システム情報科学研究院, Graduate School of Information Science and Electrical Engineering, Kyushu University

[1] http://www.wipo.int/classifications/en/

[2] http://www.nlm.nih.gov/mesh/

[3] http://www.dmoz.org/

スクによって，各文書に 1 個とする場合と，複数とする場合があるが，本稿では複数ラベル分類に取り組む．

階層的分類における興味の中心は，あらかじめ定義されたラベル階層をどのように自動分類に利用するかである．そもそも，大量のデータを階層的に組織化するという営みは，科学以前から人類が広く行なってきた．例えば，伝統社会における生物の分類もその一例である．そこでは分類の数に上限があることが知られており，その制限は人間の記憶容量に起因する可能性が指摘されている (Berlin 1992)．階層が人間の制約の産物だとすると，そのような制約を持たない計算機にとって，階層は不要ではないかと思われるかもしれない．

階層的分類におけるラベル階層の利用という観点から既存手法を整理すると，まず，非階層型と階層型に分けられる．非階層型はラベル階層を利用しない手法であり，各ラベル候補について，入力文書が所属するか否かを独立に分類する．

ラベル階層を利用する階層型は，さらに 2 種類に分類できる．一つはラベル階層を候補の枝刈りに用いる手法（枝刈り型）である．典型的には，階層を上から下にたどりながら局所的な分類を繰り返す (Montejo-Ráez and Ureña-López 2006; Qiu, Gao, and Huang 2009; Wang, Zhao, and Lu 2011)．枝刈りにより分類の実行速度をあげることができるため，ラベル階層が巨大な場合に有効である．しかし，局所的な分類を繰り返すことで誤り伝播が起きるため，精度が低下しがちという欠点が知られている (Bennett and Nguyen 2009)．もう一つの手法はパラメータ共有型である．この手法では，ラベル階層上で近いラベル同士は似通っているので，それらを独立に分類するのではなく，分類器のパラメータをラベル階層に応じて部分的に共有させる (Qiu et al. 2009)．これにより分類精度の向上を期待する．

これらの既存手法は，いずれも複数ラベル分類というタスクの特徴を活かしていない．複数ラベル分類では，最適な候補を 1 個採用すればよい単一ラベル分類と異なり，ラベルをいくつ採用するかの加減が人間作業者にとっても難しい．我々は，人間作業者が出力ラベル数を加減する際，ラベル階層を参照しているのではないかと推測する．例えば，科学技術文献を分類する際，ある入力文書が林業における環境問題を扱っていたとする．この文書に対して，「林業政策」と「林業一般」という 2 個のラベルは，それぞれ単独でみると，いずれもふさわしそうである．しかし，両者を採用するのは内容的に冗長であり，よりふさわしい「林業政策」だけを採用するといった判断を人間作業者はしているかもしれない．一方，別のラベル「環境問題」は「林業政策」と内容的に競合せず，両方を採用するのが適切を判断できる．この 2 つの異なる判断は，ラベル階層に対応している．「林業政策」と「林業一般」は最下位層において兄弟関係にある一方，「林業政策」と「環境問題」はそれぞれ「農林水産」と「環境工学」という異なる大分類に属している．

このように，我々は，出力すべき複数ラベルの間にはラベル階層に基づく依存関係があると仮定する．そして，計算機に人間作業者の癖を模倣させることによって，（それが真に良い分類

であるかは別として）人間作業者の分類を正解としたときの精度が向上することを期待する．

本稿では，このような期待に基づき，ラベル間依存を利用する具体的な手法を提案する．まずは階層型複数ラベル文書分類を構造推定問題として定式化し，複数のラベルを同時に出力する大域モデルと，動的計画法による厳密解の探索手法を提案する．次に，ラベル間依存を表現する枝分かれ特徴量を導入する．この特徴量は動的計画法による探索が維持できるように設計されている．実験では，ラベル間依存の特徴量の導入により，精度の向上とともに，モデルの大きさの削減が確認された．

本稿では，2 節で問題を定義したうえで，3 節で提案手法を説明する．4 節で実験結果を報告する．5 節で関連研究に言及し，6 節でまとめと今後の課題を述べる．

2　問題設定

階層型複数ラベル文書分類では，与えられた文書に対して，それをもっともよく表すラベルの集合 $\mathcal{M} \subset \mathcal{L}$ を返す．ここで，$\mathcal{L}$ はあらかじめ定義されたラベルの集合である．$\mathcal{L}$ は図 1 のように木構造で組織化されているとする[4]．また，付与対象のラベルは葉のみであり，内部ノードはラベルとならないとする．図 1 の場合，AA，AB，BA および BB がラベル候補となる．

いくつかの記法を整理しておく．$\mathrm{leaves}(c)$ は，c の子孫である葉の集合を返す．例えば，$\mathrm{leaves}(\mathrm{A}) = \{\mathrm{AA}, \mathrm{AB}\}$．ただし，$c$ 自身が葉の場合は，$\mathrm{leaves}(c) = \{c\}$．$p \to c$ は親 p から子 c への辺を表す．$\mathrm{path}(c)$ は ROOT と c を結ぶ辺の集合を返す．例えば，$\mathrm{path}(\mathrm{AB}) = \{\mathrm{ROOT} \to \mathrm{A}, \mathrm{A} \to \mathrm{AB}\}$．また，$\mathrm{tree}(\mathcal{M}) = \bigcup_{l \in \mathcal{M}} \mathrm{path}(l)$ とする．これは $\mathcal{M}$ を被覆する最小の部分木に対応する．例えば，$\mathrm{tree}(\{\mathrm{AA}, \mathrm{AB}\}) = \{\mathrm{ROOT} \to \mathrm{A}, \mathrm{A} \to \mathrm{AA}, \mathrm{A} \to \mathrm{AB}\}$．

文書 x は $\phi(x)$ により特徴量ベクトルに変換される．特徴量として，例えば，文書分類タスク

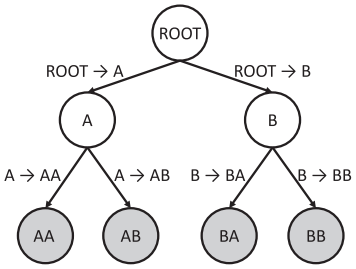


図 1　ラベル階層の例（灰色の葉のみが付与対象のラベル）

[4] いくつかの既存研究では，有向非循環グラフ (directed acyclic graph, DAG) を扱っている (Labrou and Finin 1999; LSHTC3 2012)．有向非循環グラフでは，木と異なり，ノードが一般に複数の親を持ち得る．有向非循環グラフへの対応は今後の課題とし，本稿では木構造に対象をしぼる．

で一般な単語かばん (bag-of-words) 手法を用いることができる．

本タスクは教師あり設定であり，訓練データ $\mathcal{T} = \{(x_i, \mathcal{M}_i)\}_{i=1}^{T}$ が与えられる．$\mathcal{T}$ を用いてモデルを訓練し，これとは別のテストデータによって性能を評価する．

3 提案手法

3.1 大域モデル

ラベル間依存を利用するための準備として，入力文書 x に対して出力ラベル集合 $\mathcal{M}$ を同時に推定する大域モデルを提案する．具体的には，階層的複数ラベル文書分類を構造推定問題とみなし，$\mathcal{M}$ が作る部分木に対してスコアを定義する．

$$\mathrm{score}^{\mathrm{global}}(x, \mathcal{M}) = \mathbf{w}^{\mathrm{global}} \cdot \Phi^{\mathrm{global}}(x, \mathrm{tree}(\mathcal{M}))$$

$\mathbf{w}^{\mathrm{global}}$ は重みベクトルであり，訓練データを用いて学習すべきパラメータである．$\mathbf{w}^{\mathrm{global}}$ は，辺に対応する局所的な重みベクトルを連結することにより構成される．例えば，図 1 の場合は

$$\mathbf{w}^{\mathrm{global}} = \mathbf{w}_{\mathrm{ROOT}\to\mathrm{A}} \oplus \mathbf{w}_{\mathrm{ROOT}\to\mathrm{B}} \oplus \mathbf{w}_{\mathrm{A}\to\mathrm{AA}} \oplus \mathbf{w}_{\mathrm{A}\to\mathrm{AB}} \oplus \mathbf{w}_{\mathrm{B}\to\mathrm{BA}} \oplus \mathbf{w}_{\mathrm{B}\to\mathrm{BB}}$$

となる．特徴関数 Φ^{global} は，文書 x と $\mathrm{tree}(\mathcal{M})$ を入力とし，$\mathbf{w}^{\mathrm{global}}$ と同次元のベクトルを返す．具体的には，各 $p \to c \in \mathrm{tree}(\mathcal{M})$ に対応する部分ベクトルに $\phi(x)$ を，残りの要素に 0 を入れた特徴量ベクトルを返す．したがって，$\mathrm{score}^{\mathrm{global}}(x, \mathcal{M})$ は以下のように書き換えられる．

$$\mathrm{score}^{\mathrm{global}}(x, \mathcal{M}) = \sum_{p\to c \in \mathrm{tree}(\mathcal{M})} \mathbf{w}_{p\to c} \cdot \phi(x)$$

この定式化により，$\mathbf{w}^{\mathrm{global}}$ が与えられた時，部分木のスコアを最大化する $\mathcal{M}$ を探す問題となる．

$$\operatorname*{argmax}_{\mathcal{M}} \quad \mathrm{score}^{\mathrm{global}}(x, \mathcal{M})$$

3.2 動的計画法による解探索

大域モデルの，現在のパラメータ $\mathbf{w}^{\mathrm{global}}$ のもとでの厳密解は，動的計画法により効率的に求められる．Algorithm 1 に動的計画法の擬似コードを示す．$\textsc{maxtree}(x, p)$ は，p を根とする部分木の集合から，スコアが最大のものを再帰的に探索する．したがって，我々が呼び出すのは $\textsc{maxtree}(x, \mathrm{ROOT})$ である．子 c は，(1) c を根とするスコア最大の部分木を作るラベル集合，および (2) そのスコアとひも付けされている．ただし，葉のスコアは 0 である．p から見た c のスコアは，c の部分木のスコアと辺 $p \to c$ のスコアの和である（3–8 行目）．

p の部分木のスコアを最大にするには，正のスコアを持つ c をすべて採用すればよい（10 行

Algorithm 1 MAXTREE(x, p)

入力： 文書 x, 木のノード p
出力： ラベル集合 $\mathcal{M}$, スコア s

1: $\mathcal{U} \leftarrow \{\}$
2: **for all** p の各子 c **do**
3:　**if** c が葉 **then**
4:　　$\mathcal{U} \leftarrow \mathcal{U} \cup \{(\{c\}, \mathbf{w}_{p\to c} \cdot \phi(x))\}$
5:　**else**
6:　　$(\mathcal{M}', s') \leftarrow$ MAXTREE(x, c)
7:　　$\mathcal{U} \leftarrow \mathcal{U} \cup \{(\mathcal{M}', s' + \mathbf{w}_{p\to c} \cdot \phi(x))\}$
8:　**end if**
9: **end for**
10: $\mathcal{R} \leftarrow \{(\mathcal{M}', s') \in \mathcal{U} | s' > 0\}$
11: **if** $\mathcal{R}$ が空 **then**
12:　$\mathcal{R} \leftarrow \{(\mathcal{M}', s')\}$ ただし，$(\mathcal{M}', s')$ は $\mathcal{U}$ のなかで s' が最大のもの
13: **end if**
14: $\mathcal{M} \leftarrow \bigcup_{(\mathcal{M}', s') \in \mathcal{R}} \mathcal{M}'$
15: $s \leftarrow \sum_{(\mathcal{M}', s') \in \mathcal{R}} s'$
16: **return** $(\mathcal{M}, s)$

目）．いずれの子も正のスコアを持たない場合は，最大のスコアを持つ子を 1 個採用する（11–13 行目）．採用された子の集合により，p のラベル集合とスコアが決定される（14–15 行目）．

このアルゴリズムの拡張としては，上位 N 個の候補集合を出すというものが考えられる．木に対する動的計画法としては，構文解析 (McDonald, Crammer, and Pereira 2005) よりもはるかに簡単なため，上位 N 個への拡張 (Collins and Koo 2005) もさほど難しくない．

3.3　ラベル間依存の利用

以上の準備により，ラベル間依存を利用する条件が整った．ラベル間依存の捕捉は，大域モデルに対する特徴量の追加により実現される．具体的には，あるノードがいくつの子を採用しやすいかを制御する枝分かれ特徴量を導入する．

枝分かれ特徴量は $\phi^{\mathrm{BF}}(p, k)$ により表される．ここで p は根あるいは内部ノードであり，k は p が採用する子の数である．ただし，あらゆる k の値に対して特徴量を設けると疎になるため，ある R について，$R+1$ 個 $(1, \cdots, R$ もしくは $> R)$ の特徴量に限定する．さらに，ノードごとの特徴量だけでなく，すべての根あるいは内部のノードが共有する $R+1$ 個の特徴量も設ける．つまり，追加される特徴量は $(I+1)(R+1)$ 個であり，各ノードに対して 2 個の特徴量が発火する．ここで，I はラベル階層における根および内部ノードの個数とする．

この枝分かれ特徴量は，動的計画法による厳密解探索が維持できるように設計されている．この特徴量を組み込むには，Algorithm 1 の 10–15 行目を Algorithm 2 で置き換えればよい．枝分かれ特徴量のスコア $\mathbf{w}^{\mathrm{BF}} \cdot \phi^{\mathrm{BF}}(p, k)$ は k のみに依存する．そこで，まずは採用する子の数 k

Algorithm 2 枝分かれ特徴量を組み込むための修正（Algorithm 1 の 10–15 行目を以下で置き換える）

10: $r \leftarrow \mathcal{U}$ を s により降順にソートした配列
11: $\mathcal{R}' \leftarrow \{\}, \quad s' \leftarrow 0, \quad \mathcal{M}' \leftarrow \{\}$
12: **for** $k = 1..$size of r **do**
13: $\quad (\mathcal{M}, s) \leftarrow r[k]$
14: $\quad s' \leftarrow s' + s, \quad \mathcal{M}' \leftarrow \mathcal{M}' \cup \mathcal{M}$
15: $\quad \mathcal{R}' \leftarrow \mathcal{R}' \cup \{(\mathcal{M}', s' + \mathbf{w}^{\mathrm{BF}} \cdot \phi^{\mathrm{BF}}(p, k))\}$
16: **end for**
17: $(\mathcal{M}, s) \leftarrow \mathcal{R}'$ のなかでスコア s が最大の要素

によって候補をグループ分けし，各グループのなかでスコアが最大の候補を選ぶ（12–16 行目）．最後に，異なるグループ同士を比較し，スコアが最大となる候補を採用する（17 行目）．グループ内でスコアが最大の候補を選ぶには，子をスコア順に並べ，上位 k 個を採用すれば良い．候補のスコアは，p から見た各子のスコアと枝分かれ特徴量のスコアの和となる（15 行目）．

枝分かれ特徴量の導入により，ラベルの採否の判断が，ラベル同士の相対的な比較によって行われるようになる．1 節で触れた，「林業政策」と「環境問題」というラベルが付与された文書を再び例に挙げる．この文書に対して「林業一般」というラベルはそれほど不適切には見えないが，枝分かれ特徴量を持たないモデルは，「林業一般」を付与**しない**理由を，$\phi(x)$ に対応する重みですべて説明しなければならない．4.5 節で示すように，枝分かれ特徴量の重みは，一般に，負の値を持ち，ペナルティとして働く．また，子の数が増えるにつれてペナルティが増えるように学習される．したがって，子を 2 個採用するとよりペナルティがかかるので，「林業一般」に対応する重みを無理に引き下げることなく，相対的により適切な「林業政策」のみを採用することが可能となる．

3.4 大域訓練

大域モデルの訓練手法をここでは大域訓練と呼ぶ．本稿では，パーセプトロン系のオンライン学習アルゴリズムを採用する．具体的には，構造推定問題に対する Passive-Aggressive アルゴリズム (Crammer, Dekel, Keshet, Shalev-Shwartz, and Singer 2006) を用いる．Passive-Aggressive を採用した理由としては，実装の簡便さ，バッチ学習と異なり，大量の訓練データに容易に対応可能なオンライン学習であること，次節で述べるように並列分散化が容易に実現できることが挙げられる．ただし，これは提案手法がパーセプトロン系アルゴリズムでしか実現できないことを意味せず，構造化 SVM (Tsochantaridis, Hofmann, Joachims, and Altun 2004) を含む他の構造学習アルゴリズムの導入も検討に値する．

大域モデルの場合の擬似コードを Algorithm 3 に示す．ここで，N は訓練の反復数を表し，パラメータ C は 1.0 とする．現在のパラメータにおける厳密解は上述の動的計画法により求ま

Algorithm 3 大域訓練のための Passive-Aggressive アルゴリズム（PA-I，予測ベース更新）

入力： 訓練データ $\mathcal{T} = \{(x_i, \mathcal{M}_i)\}_{i=1}^{T}$
出力： 重みベクトル $\mathbf{w}^{\mathrm{global}}$

1: $\mathbf{w}^{\mathrm{global}} \leftarrow \mathbf{0}$
2: **for** $n = 1..N$ **do**
3:　$\mathcal{T}$ をシャッフル
4:　**for all** $(x, \mathcal{M}) \in \mathcal{T}$ **do**
5:　$\hat{\mathcal{M}} \leftarrow \mathrm{argmax}_{\mathcal{M}}\ \mathrm{score}^{\mathrm{global}}(x, \mathcal{M})$
6:　$\rho \leftarrow 1 - 2|\mathcal{M} \cap \hat{\mathcal{M}}|/(|\mathcal{M}| + |\hat{\mathcal{M}}|)$
7:　**if** $\rho > 0$ **then**
8:　$l \leftarrow \mathrm{score}^{\mathrm{global}}(x, \hat{\mathcal{M}}) - \mathrm{score}^{\mathrm{global}}(x, \mathcal{M}) + \sqrt{\rho}$
9:　$\tau \leftarrow \min\{C, \frac{l}{\|\Phi^{\mathrm{global}}(x, \mathrm{tree}(\mathcal{M})) - \Phi^{\mathrm{global}}(x, \mathrm{tree}(\hat{\mathcal{M}}))\|^2}\}$
10:　$\mathbf{w}^{\mathrm{global}} \leftarrow \mathbf{w}^{\mathrm{global}} + \tau(\Phi^{\mathrm{global}}(x, \mathrm{tree}(\mathcal{M})) - \Phi^{\mathrm{global}}(x, \mathrm{tree}(\hat{\mathcal{M}})))$
11:　**end if**
12:　**end for**
13: **end for**

る（5 行目）．予測を誤った場合，正解ラベル集合を出力する方向に重みを更新する（10 行目）．ここで，コスト ρ はモデル予測の誤り度合いを表し，重みの更新幅を変化させる．ρ は，正解ラベル集合とシステムの出力の一致の度合いに基づいている．

3.5　大域訓練の並列分散化

大域訓練には学習が非常に遅いという欠点がある．ラベル集合の分類はラベル 1 個の 2 値分類とは比較にならないほど遅い．しかも，大域訓練はモデルを一枚岩とするため，モデルを局所分類器に分割して並列化することができない．

そこで，繰り返しパラメータ混ぜ合わせ法 (McDonald, Hall, and Mann 2010) を用いて並列分散化を行う．基本的な考えは，モデルを分割する代わりに，訓練データを分割することで並列化を行うというものである．別々の訓練データ断片から学習されたモデル群を繰り返し混ぜ合わせることで収束性を保証している．

Algorithm 4 に繰り返しパラメータ混ぜ合わせ法の擬似コードを示す．ここで N' は繰り返しパラメータ混ぜ合わせ法の反復数，S は訓練データの分割数を表す．繰り返しパラメータ混ぜ合わせ法では，断片ごとに並列に訓練を行う．各反復の最後に，並列に訓練された複数のモデルを平均化する．次の反復では，この平均化されたモデルを初期値として用いる．

繰り返しパラメータ混ぜ合わせ法はパーセプトロン向けに提案されたものである．しかし，(McDonald et al. 2010) が言及している通り，Passive-Aggressive アルゴリズムに対しても収束性を証明することができる．

Algorithm 4 繰り返しパラメータ混ぜ合わせ法による大域訓練

入力： 訓練データ $\mathcal{T} = \{(x_i, \mathcal{M}_i)\}_{i=1}^{T}$
出力： 重みベクトル $\mathbf{w}^{\text{global}}$

```
T を T_1, ··· T_S に分割
w^global ← 0
for n = 1..N' do
    for s = 1..S do
        w_s^global ← 非同期的に Algorithm 3 を呼び出す．ただしいくつかの修正を加える．T を T_s で置き換える．w^global を 0 ではなく w^global で初期化する．反復数を N = 1 とする．
    end for
    非同期処理の終了を待つ
    w^global ← (1/S) Σ_{s=1}^{S} w_s^global
end for
```

ジャーナル名	Journal of Wood Science
標題	京都約束期間後の収穫木材生産量算出
抄録	持続可能な林業からの収穫木材製品（ＨＷＰ）量の増大は，大気中炭素レベル低下の支援になる可能性がある．京都議定書の（２００８～２０１２の）最初の約束期間では，ＨＷＰのこの炭素総量効果は無視されており，森林収穫量は二酸化炭素の瞬間排出量として処理されている．しかし，国連気候変動枠組条約会議の２０１３年からの次の約束期間では，ＨＷＰの結果である炭素総量変化には国の温室ガスインベントリーが考慮されるだろう．日本木材学会は，木材連合に加盟する８か国の学会，工業協会および非政府組織の円卓会議への出席を要請された．その会議では，ＨＷＰに対する算出法が検討され，次期約束期間には総量変化法を採用すべきであるという合意に達した．
分類コード	FF01020X (林業政策), SA01020V (環境問題)

図 2　JSTPlus の文書例

4　実験

4.1　データ

評価データとして JSTPlus[5]を用いる．JSTPlus は科学技術振興機構が作成している科学技術文献のデータベースである．各文書は，標題，抄録，著者一覧，ジャーナル名，分類コード一覧や，その他数多くの項目からなる．文書例を図 2 に示す．実験では，標題と抄録を文書分類に用いるテキストとし，分類コードを付与すべきラベルとみなす．また，2010 年の文献のうち，

[5] http://jdream3.com/service/jdream.html

日本語の標題と日本語の抄録の両方を含むものを実験の対象とした．その結果，455,311 件の文書を得た．これを 409,892 件の訓練データと 45,419 件の評価データに分割した．

ラベル（分類コード）は 3,209 個からなり，これは 4,030 個の辺に対応する．ラベル階層は，根を除いて，最大で 5 階層となっている．ただし，いくつかの辺は中間層を飛ばす（例えば，第 2 層のノードの子が第 4 層にある場合がある）．各文書は平均で 1.85 個のラベルが付与されている（分散は 0.85）．文書ごとの最大ラベル数は 9 である．

文書の特徴関数 $\phi(x)$ には以下の 2 種類の特徴量を用いる．

(1) ジャーナル名．2 値特徴量で，各文書につき 1 個の特徴量が発火する．

(2) 標題と抄録中の内容語．値は頻度．ただし，標題中の内容語の頻度は 2 倍する．

内容語抽出には，形態素解析器 JUMAN[6]および構文解析器 KNP[7]を用いた．まず JUMAN によって各文を単語列に分割し，次に KNP が持つ規則を使って内容語にタグ付けした．各文書は平均で 380 文字を含んでいた．これは内容語としては 120 語に相当する．

4.2 モデル設定

大域訓練で訓練された大域モデル (**GM-GT**) について，枝分かれ特徴量 (**BF**) を用いた場合と用いなかった場合を比較する．大域モデルの繰り返しパラメータ混ぜ合わせ法については，訓練データを 10 個の断片に分割し，反復数は $N' = 10$ とする．枝分かれ特徴量について，$R = 3$ とする．

その他の比較対象として，従来研究を参考にして以下のモデルを用いる．

4.2.1 非階層型

非階層型 (**FLAT**) はラベル階層を無視し，各ラベル l を文書 x に付与すべきか否かを独立に決定する．そのために各 l に対して 2 値分類器を用意する．分類器の実装手法としては，ナイーブベイズ，ロジスティック回帰，サポートベクタマシンなどが用いられてきたが，本稿では，提案手法との比較のために Passive-Aggressive アルゴリズム (Crammer et al. 2006) を用いる．

ラベル l に対する 2 値分類器は重みベクトル $\mathbf{w}_l$ を持つ．スコア $\mathbf{w}_l \cdot \phi(x)$ が正のとき，l を x に付与する．ただし，文書に対して最低 1 個のラベルを付与する．そのために，いずれのラベルも正のスコアを取らない場合は，一番高いスコアを持つラベルを 1 個採用する．

$\mathbf{w}_l$ を訓練するために，元の訓練データ $\mathcal{T}$ を以下のようにして $\mathcal{T}_l$ に変換する．

$$\mathcal{T}_l = \left\{ (x_i, y_i) \;\middle|\; \begin{array}{ll} y_i = +1 & \text{if } l \in \mathcal{M}_i \\ y_i = -1 & \text{otherwise} \end{array} \right\}_{i=1}^{T}$$

[6] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[7] http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

Algorithm 5 2 値分類器に対する Passive-Aggressive アルゴリズム (PA-I)

入力： 訓練データ $\mathcal{T}_l = \{(x_i, y_i)\}_{i=1}^{T}$
出力： 重みベクトル $\mathbf{w}_l$

```
1: w_l ← 0
2: for n = 1..N do
3:     T_l をシャッフル
4:     for all (x, y) ∈ T_l do
5:         l ← max{0, 1 − y(w_l · φ(x))}
6:         if l > 0 then
7:             τ ← min{C, l / ||φ(x)||^2}
8:             w_l ← w_l + τ y φ(x)
9:         end if
10:    end for
11: end for
```

各文書はラベル l を持つとき正例，そうでなければ負例となる．擬似コードを Algorithm 5 に示す．ここで，パラメータ C は 1.0 とする．訓練の反復数は $N = 10$ とする．なお，各 2 値分類器は独立なので，訓練は容易に並列化できる．

4.2.2 枝刈り型

枝刈り型 (**PRUNE**) はラベル階層を利用する手法であり，ラベル階層に対応する 2 値分類器の集合を持つ (Montejo-Ráez and Ureña-López 2006; Wang et al. 2011; Sasaki and Weissenbacher 2012)[8]．各 2 値分類器はラベル階層上の辺 $p \to c$ とひも付けされ，重み $\mathbf{w}_{p\to c}$ を持つ．$\mathbf{w}_{p\to c} \cdot \phi(x) > 0$ は，x を p のいずれかの子孫に割り当てるべきであることを表す．これらの 2 値分類器も並列に訓練できる．パラメータ C の値，訓練の反復数は非階層型と同じとする．

枝刈り型には誤り伝播 (Bennett and Nguyen 2009) とよばれる問題が知られている．すなわち，階層上位の分類器による誤りから回復する手段がないため，累積的に誤りが作用する．誤り伝播を軽減するために様々な手法が提案されているが，煩雑さを避けるため，本稿では，Algorithm 6 に示す単純な実装を採用する．各ノード p において，局所分類器が正のスコアを返す子すべてを採用する（4–7 行目）．ただし，いずれの子も正のスコアを得ない場合は，一番高いスコアを得た子を 1 つ採用する（8–10 行目）．この操作を葉に到達するまで繰り返す．

2 値分類器の訓練データ $\mathcal{T}_{p\to c}$ の構築方法としては，以下の 2 種類を試す．

ALL 全訓練データを利用する (Punera and Ghosh 2008).

$$\mathcal{T}_{p\to c} = \left\{ (x_i, y_i) \;\middle|\; \begin{array}{ll} y_i = +1 & \text{if } \exists l \in \mathcal{M}_i, l \in \text{leaves}(c) \\ y_i = -1 & \text{otherwise} \end{array} \right\}_{i=1}^{T}$$

[8] ラベル階層に対応する 2 値分類器の集合を使う他の手法としては，(Punera and Ghosh 2008) が階層上の等調回帰により，局所分類器の出力を後処理している．

Algorithm 6 枝刈り型探索

入力： 文書 x
出力： ラベル集合 $\mathcal{M}$

$q \leftarrow$ [ROOT], $\mathcal{M} \leftarrow \{\}$
while q が空でない **do**
  $p \leftarrow q$ の最初の要素を取り出す, $\mathbf{t} \leftarrow \{\}$
  for all p の子である c **do**
    $\mathbf{t} \leftarrow \mathbf{t} \cup \{(c, \mathbf{w}_{p\to c} \cdot \phi(x))\}$
  end for
  $\mathcal{U} \leftarrow \{(c, s) \in \mathbf{t} | s > 0\}$
  if $\mathcal{U}$ が空 **then**
    $\mathcal{U} \leftarrow \{(c, s)\}$, ただし c は p の子のなかで一番高いスコア s を持つ
  end if
  for all $(c, s) \in \mathcal{U}$ **do**
    if c が葉 **then**
      $\mathcal{M} \leftarrow \mathcal{M} \cup \{c\}$
    else
      c を q に追加
    end if
  end for
end while

各文書は c のいずれかの子孫のラベルが割り当てられていれば正例，そうでなければ負例となる．

SIB　正例は **ALL** と同じだが，負例を c の兄弟の子孫が割り当てられている場合に限定する．

$$\mathcal{T}_{p\to c} = \left\{ (x, y) \;\middle|\; \begin{array}{ll} y = +1 & \text{if } \exists l \in \mathcal{M}, l \in \text{leaves}(c) \\ y = -1 & \text{if } \exists l \in \mathcal{M}, l \in \text{leaves}(p) \text{ かつ } l \notin \text{leaves}(c) \end{array} \right\}$$

こうすることで，全体として小さなモデルが学習される．なぜなら，数の多い階層下位の分類器に与えられる訓練データが小さくなるからである．従来研究では **SIB** を採用する場合が多い (Liu, Yang, Wan, Zeng, Chen, and Ma 2005; Wang et al. 2011; Sasaki and Weissenbacher 2012).

4.3　評価尺度

複数ラベル分類に対する評価尺度は数多く存在するが，大きく 2 種類に整理できる．1 つは，文書を単位とした評価尺度で，しばしば用例ベースの尺度とよばれる (Godbole and Sarawagi 2004; Tsoumakas, Katakis, and Vlahavas 2010)．文書単位の尺度として，適合率 (EBP)，再現率 (EBR) および F 値 (EBF) が以下のように定義される．

$$\text{EBP} = \frac{1}{T} \sum_{i=1}^{T} \frac{|\mathcal{M}_i \cap \hat{\mathcal{M}}_i|}{|\hat{\mathcal{M}}_i|}$$

$$\text{EBR} = \frac{1}{T} \sum_{i=1}^{T} \frac{|\mathcal{M}_i \cap \hat{\mathcal{M}}_i|}{|\mathcal{M}_i|}$$

$$\mathrm{EBF} = \frac{1}{T}\sum_{i=1}^{T}\frac{2|\mathcal{M}_i \cap \hat{\mathcal{M}}_i|}{|\hat{\mathcal{M}}_i| + |\mathcal{M}_i|}$$

ここで T はテストデータ中の文書数，$\mathcal{M}_i$ は i 番目の文書の正解ラベル集合，$\hat{\mathcal{M}}_i$ はそれに対応するシステムの出力とする．

もう一つは，ラベルを単位とした評価尺度で，通常の適合率，再現率および F 値が用いられる．ただし，複数のラベルの集計方法としてマクロ平均とマイクロ平均がある (Tsoumakas et al. 2010)．そのため合計で，LBMaP，LBMaR，LBMaF，LBMiP，LBMiR および LBMiF の 6 種類の尺度を用いる．

最後に階層的な評価も行う (Kiritchenko 2005)．これは，出力ラベルがラベル階層上において正解と近いときに「部分点」を与えるものである．今回のように循環がない木構造を仮定した場合，適合率 (hP) および再現率 (hR) は以下のように定義される．

$$\mathrm{hP} = \frac{\sum_{i=1}^{T}|\mathrm{tree}(\mathcal{M}_i) \cap \mathrm{tree}(\hat{\mathcal{M}}_i)|}{\sum_{i=1}^{T}|\mathrm{tree}(\hat{\mathcal{M}}_i)|}$$

$$\mathrm{hR} = \frac{\sum_{i=1}^{T}|\mathrm{tree}(\mathcal{M}_i) \cap \mathrm{tree}(\hat{\mathcal{M}}_i)|}{\sum_{i=1}^{T}|\mathrm{tree}(\mathcal{M}_i)|}$$

F 値 (hF) は hP と hR の調和平均として定義される．

4.4 結果

各種モデルの精度比較を表 1 に示す．枝分かれ特徴量を組み込んだ大域モデル (**GM-GT-BF**) が 7 種類の尺度で最高精度を得た．枝分かれ特徴量なしのモデル (**GM-GT**) と比較する

表 1 モデルの比較結果

モデル	反復数	時間（分）	大きさ	EBP	EBR	EBF
GM-GT	10	310	68M	**.5177**	.4096	.4317
GM-GT-BF	10	315	62M	.5172	.4121	**.4347**
FLAT	10	266	73M	.4520	.4111	.3956
PRUNE-ALL	10	**5**	115M	.3927	.4064	.3713
PRUNE-SIB	10	5	**39M**	.4010	**.4396**	.3881

モデル	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF	hP	hR	hF
GM-GT	.4301	.2708	.2659	.5085	.3655	.4253	.6458	.4843	.5535
GM-GT-BF	**.4519**	.2645	**.2709**	**.5132**	.3701	**.4300**	**.6493**	.4898	**.5584**
FLAT	.4260	.2549	.2578	.4155	.3727	.3930	.5343	.4746	.5027
PRUNE-ALL	.3288	.2764	.2415	.3622	.3716	.3668	.4988	.5060	.5024
PRUNE-SIB	.3291	**.2989**	.2515	.3415	**.4066**	.3712	.4750	**.5359**	.5036

と，EBP，LBMaR 以外の尺度で **GM-GT-BF** が上回り，すべての F 値を改善した．この改善は統計的に有意 ($p < 0.01$) であった．

大域モデルを非階層型 (**FLAT**) と比較すると，適合率の改善が著しい一方，再現率に大きな差は見られない．2 種類の枝刈り型 (**PRUNE**) を比較すると，兄弟のみで訓練する場合 (**SIB**) の方が全体的にやや良い精度が得られた．しかし，多くの尺度で非階層型に敗れており，従来研究の結果を再現する形となっている．

誤り例を見ると，誤って採用したラベル，誤って採用しなかったラベルのいずれも，正解ラベルから離れて人間として改めて判断すると，必ずしも誤りとは言い切れない場合が少なくなかった．特に，該当文書にとって周辺的な話題を表すラベルをどこまで採用べきかを判断するのが難しかった．なお，モデル間の分類結果の差分からは，明確な誤り，改善の傾向をつかむのは困難であった．

時間はテストデータの分類に要した時間であり，モデルの読み込み時間は含まない[9]．予想される通り，枝刈り型が圧倒的に速い．**GM-GT-BF** は **PRUNE-ALL** と比較して約 60 倍の時間を要した．しかし，**FLAT** と比較すると，階層を利用するにも関わらず，約 18%の増加にとどまっている．これは，**GM-GT-BF** のモデルの大きさが **FLAT** よりも約 16%小さいことで説明できるかもしれない．

モデルの大きさは重みベクトル中で，絶対値が 10^{-7} より大きい要素の数とする．大きさは **PRUNE-SIB** が最小で，**PRUNE-ALL** が最大となった．**GM-GT-BF** が **GM-GT** よりも大きさを約 9%削減したことは特筆に値する．訓練に用いた Passive-Aggressive アルゴリズムには重みを 0 につぶそうとする仕組みがないことから，大きさが削減された理由は，学習過程で **GM-GT-BF** が **GM-GT** よりも予測を誤る回数が少なかったからと考えられる．このように，より小さなモデルでより高い精度が得られたことは，出力すべき複数ラベルの間にはラベル階層に基づく依存関係があるという我々の仮定を支持するものと考える．

4.5　議論

大域モデルの重み $\mathbf{w}^{\mathrm{global}}$ 自体は，大域訓練 (**GT**) だけでなく，枝刈り型で用いた 2 値分類器群を連結することによっても構成できる (**LT**)．大域モデルの性質をさらに調べるために，こうしたモデルとの比較も行った．

表 2 に大域モデルの訓練方法の比較結果を示す．訓練データとして **SIB** を用いた場合，極端に多くの候補を出力するようになり，その結果，極端に低い適合率と高い再現率を得た．**SIB** という限定されたデータで訓練された局所的な分類器に対して，大域モデルが未知の文書の分類を行わせたため，このような不安定な振る舞いとなった．一方，訓練データとして **ALL** を用

[9] 実験では 8 コア Intel Xeon 2.70 GHz CPU の 1 コア，64 GB のメモリを用い，実装には Perl を用いた．

いた場合，枝刈り型 (**PRUNE-ALL**) から精度を大幅に向上させ，大域訓練とくらべても遜色のない精度が得られた．モデルの大きさや分類速度において大域訓練に劣りはするものの，大域モデルの最適化を行わずにこのような高精度が得られたことは興味深い．これは，訓練手法に改善の余地があることを示唆する．本稿では 10 並列による繰り返しパラメータ混ぜ合わせ法を用いたが，今後の最適化技術の発展が期待される．

表 3 に訓練データに対する精度を示す．訓練データに対しては非階層型 (**FLAT**) が一番高い精度を示し，**ALL** により局所訓練された大域モデル (**GM-LT-ALL**) がそれに続いた．大域訓練を行った場合 (**GM-GT-BF**) との比較から，局所訓練が過学習をもたらしているとみられる．

また，局所訓練と大域モデルの組み合わせにより，枝刈り型探索が誤りの主要因であることが確認できた．すなわち，**PRUNE-ALL** を訓練データに適用したところ，33%の文書について，**PRUNE-ALL** が出力したラベル集合よりも，正解ラベル集合の方が大域モデルにおいて高いスコアを持っていた．言い換えると，正しく探索を行えば犯さない誤りであった．ただし，この高い数値には過学習の影響も含まれており，同じ操作をテストデータに適用した場合は，割合は 14%に下がった[10]．

表 2 大域モデルの訓練方法の比較

モデル	時間（分）	EBP	EBR	EBF
GT	310	**.5177**	.4096	.4317
GT-BF	315	.5172	.4121	**.4347**
LT-ALL	329	.4790	.4336	.4247
LT-SIB	298	.0026	**.6804**	.0481

モデル	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF	hP	hR	hF
GT	.4301	.2708	.2659	.5085	.3655	.4253	.6458	.4843	.5535
GT-BF	.4519	.2645	.2709	**.5132**	.3701	**.4300**	**.6493**	.4898	**.5584**
LT-ALL	**.4576**	.2760	**.2799**	.4542	.3933	.4216	.6020	.5163	.5559
LT-SIB	.0267	**.5214**	.0406	.0184	**.6649**	.0358	.0031	**.8104**	.0600

表 3 訓練データにおける精度

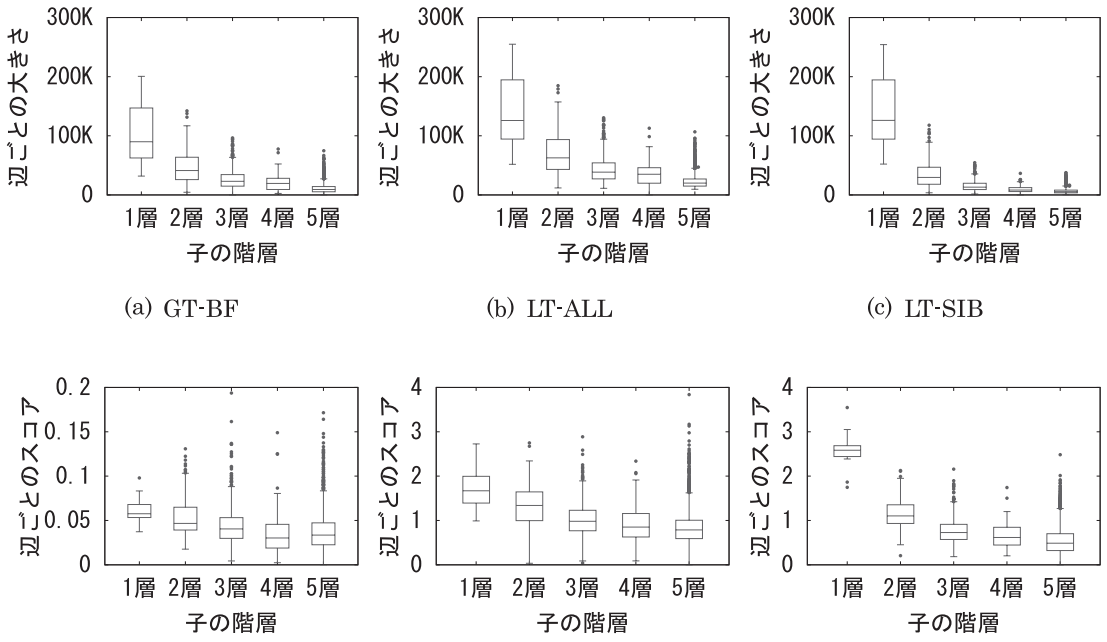
モデル	EBF	LBMiF	hF
GM-GT-BF	.7126	.6942	.7508
FLAT	.9227	.9204	.9337
GM-LT-ALL	.8977	.8951	.9114
GM-LT-SIB	.0731	.0540	.0743

[10] ラベル階層が大規模で厳密解探索が難しいといった理由で，枝刈り型探索を Algorithm 3 の訓練に用いる場合，探索誤りから生じる「非侵害」問題に対処しなければならない．すなわち，モデル予測 $\hat{\mathcal{M}}$ が正解 $\mathcal{M}$ よりも低いスコアを持つ場合，重みベクトルの更新が無効となってしまう．この問題に対処するための手法がいくつか提案されている (Collins and Roark 2004; Huang, Fayong, and Guo 2012).

より詳細にモデルを調べるために，辺に分解した結果を示す．図 3 は **GT-BF**，**LT-ALL**，**LT-SIB** の比較である．図 (a) から (c) は辺に対応する局所ベクトルの大きさを示す．ここで，大きさの定義は表 1 と同じである．辺を子の階層によって集約し，大きさを平均した結果を示す．一般に，上位階層ほど多数の有効な重みベクトルが必要となることが確認できる．**GT-BF** は **LT-ALL** よりも大きさが小さいが，辺ごとの大きさの比率は似通っている．**LT-SIB** と比較すると，**GT-BF** は上位階層では小さいが，下位階層では大きな有効重みベクトルを持つ．**LT-SIB** では兄弟からの識別のみを考慮していたが，大域学習ではすべての辺が適切なスコアを返す必要があるため，有効重みベクトルがより大きくなったとみられる．

図 (d) から (f) は，各辺が得たスコアの絶対値の平均を表す．ここで，スコアは，テストデータに対するモデル出力から計算されたものである．これにより，どの階層の辺が強くモデル出力に影響しているかが推測できる．この結果から，上位階層ほど大きな影響を持つことがわかる．しかし，**GT-BF** は他とくらべて上位階層の影響が小さい．すなわち，**GT-BF** においては下位階層の辺が相対的に重要な役割を果たしている．

枝分かれ特徴量に対応する重みを図 4 にヒートマップとして示す．各要素の値は，親ノードに与えられた重み（ノードごとの重みと共有された重みの和）を平均したものである．平均化



(a) GT-BF　(b) LT-ALL　(c) LT-SIB

(d) GT-BF　(e) LT-ALL　(f) LT-SIB

図 3　辺ごとのモデルの大きさとスコアの比較

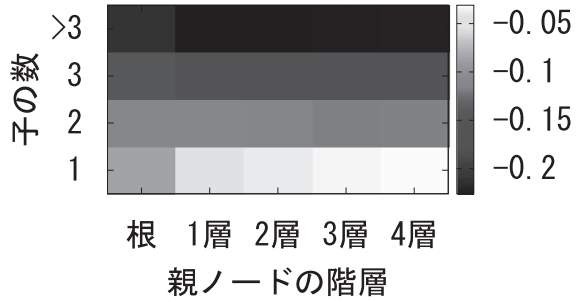


図 4　枝分かれ特徴量のヒートマップ表現

された値はすべて負となり，子の数が増えるにつれてペナルティが単調増加した．異なる階層間の重みの比較は，それらが重みベクトルの他の部分の値に依存するため難しい．しかし，下位ノードほど子の数に応じた重みの落差が大きいという結果は，階層上近いラベル候補同士ほど強い競合関係にあるという我々の仮説を支持しているようにみえる．

最後に，訓練データおよび評価データの正解ラベルについて，正解ラベルを被覆する最小の部分木を作り，親が採用する子の数を調べた．採用した子の数が複数である割合は，根で 34.9%，第 1 層で 10.1%，第 2 層で 4.6%，第 3 層で 1.5%，第 4 層で 0.6%であり，下位ノードほど強い競合関係にあることが確認できた．

5　関連研究

階層型文書分類において，枝刈り型が非階層型にしばしば敗れることが報告されており，誤り伝播を軽減するために様々な手法が提案されてきた．(Sasaki and Weissenbacher 2012) は枝刈り探索時の枝刈り基準を緩め，最後に候補の枝刈りを行う．すなわち，Algorithm 6 の 7 行目の閾値を 0 から −0.2 などに引き下げて，より多くの候補を採用する．最後に，各候補について，根から葉までのパスの（シグモイド関数で変換された）局所スコアの和を取り，これに閾値を設定することによって出力ラベルを絞り込む．S-cut (Montejo-Ráez and Ureña-López 2006; Wang et al. 2011) は，一律の閾値を用いるのではなく，局所分類器ごとに閾値を設定する手法である．R-cut は上位 r 個の候補を採用する手法で，選び方には大域的手法 (Liu et al. 2005; Montejo-Ráez and Ureña-López 2006) と局所的手法 (Wang et al. 2011) がある．(Wang et al. 2011) は採用されたラベル候補をメタ分類器にかけ，最終的な出力を決定する．メタ分類器の特徴量としては，根から葉までの局所スコアやその累積などを用いる．本稿ではこれらを総称して後付け補正とよぶ．後付け補正では，いずれもモデルあるいは探索が本質的に不完全であることを想定し，追加のパラメータによる補正を行なっている．そうしたパラメータは，人手で設定するか，あ

るいは訓練データとは別に開発データを用意して推定しなければならず煩雑である．一方，提案手法には後付け補正は不要であり，モデル自体の改善に専念できる．

ラベル階層を下から上へ探索しながら候補を探すという点で，提案手法と似た手法が (Bennett and Nguyen 2009) により提案されている．しかし，彼らの手法では，大域モデルも大域訓練も用いられていない．代わりに，階層下位の分類器のスコアが上位の分類器のメタ特徴量として用いられている．分類器の訓練は局所的に行われ，煩雑な交差確認を必要とする．

本稿ではあらかじめ定義されたラベル階層を利用した．そうした手がかりがない場合にラベル間依存を捉えるための手法も研究されている．(Ghamrawi and McCallum 2005; Miyao and Tsujii 2008) は，出力すべきラベル集合中のラベルペアを特徴量に組み込んでいる．本稿のようにラベル階層が利用できる場合は，それをもとに限られた数のラベル同士の関係を考慮すればすむ．一方，ラベル階層がない場合は，モデルはすべてのラベルペアを考慮する必要があり，訓練および解探索に大きな計算コストを要する．こうしたモデルの検証は，ラベルの異なり数が数十程度のデータセットを用いて行われてきた．ラベルの異なり数が大きな場合について，(Tai and Lin 2010) は，ラベル集合を低次元の直交座標系に写像し，この空間上で非階層型の分類器を学習する手法を提案している．予測時には，分類器の出力を元の空間へ写像するという自明でない復号が必要となる．(Bi and Kwok 2011) は，ラベル階層を組み込むために，木あるいは有向非循環グラフの制約を満たすような復号手法を提案している．

6 おわりに

本稿では，階層型複数ラベル文書分類を構造推定問題として定式化し，動的計画法による厳密解探索方法，大域訓練，ラベル間依存をとらえる枝分かれ特徴量を提案した．枝分かれ特徴量はモデルの大きさを削減するとともに精度の向上をもたらした．この結果は，人間作業者が複数のラベル候補から出力を選択する際，ラベル階層に基づいて，競合する候補の相対的な重要性を考慮していることを示唆する．

今後の方向性としては，枝分かれ特徴量以外によってラベル間依存をとらえる方法を探究するというものが考えられる．例えば，「〜その他」や「〜一般」といったラベルは，他のラベルとの関係において特殊な振る舞いをすると予想される．また，本稿では葉のみが付与対象ラベルという問題設定を行ったが，従来研究には内部ノードも付与対象である場合を扱ったものがある (Liu et al. 2005)．こうした内部ノードの振る舞いも特殊である．内部ノードを採用するとき，その子孫へのラベル付与を行わないことが多い．さらに，木構造から有向非循環グラフへの提案手法の一般化も課題である．

謝　辞

本研究で評価実験に用いた JSTPlus は，共同研究を通じて，独立行政法人科学技術振興機構に提供していただきました．深く感謝いたします．本研究は一部 JST CREST の支援を受けました．

参考文献

Bennett, P. N. and Nguyen, N. (2009). "Refined Experts: Improving Classification in Large Taxonomies." In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*, pp. 11–18.

Berlin, B. (1992). *Ethnobiological Classification: Principles of Categorization of Plants and Animals in Traditional Societies*. Princeton University Press.

Bi, W. and Kwok, J. (2011). "Multi-Label Classification on Tree- and DAG-Structured Hierarchies." In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 17–24.

Collins, M. and Koo, T. (2005). "Discriminative Reranking for Natural Language Parsing." *Computational Linguistics*, **31** (1), pp. 25–70.

Collins, M. and Roark, B. (2004). "Incremental Parsing with the Perceptron Algorithm." In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pp. 111–118.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). "Online Passive-Aggressive Algorithms." *Journal of Machine Learning Research*, **7**, pp. 551–585.

Ghamrawi, N. and McCallum, A. (2005). "Collective Multi-label Classification." In *CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pp. 195–200.

Godbole, S. and Sarawagi, S. (2004). "Discriminative Methods for Multi-labeled Classification." In Dai, H., Srikant, R., and Zhang, C. (Eds.), *Advances in Knowledge Discovery and Data Mining*, Vol. 3056 of *Lecture Notes in Computer Science*, pp. 22–30. Springer Berlin Heidelberg.

Huang, L., Fayong, S., and Guo, Y. (2012). "Structured Perceptron with Inexact Search." In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–151.

Kiritchenko, S. (2005). *Hierarchical Text Categorization and Its Application to Bioinformatics*.

Ph.D. thesis, University of Ottawa.

Labrou, Y. and Finin, T. (1999). “Yahoo! as an Ontology: Using Yahoo! Categories to Describe Documents.” In *Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM ’99)*, pp. 180–187.

Liu, T.-Y., Yang, Y., Wan, H., Zeng, H.-J., Chen, Z., and Ma, W.-Y. (2005). “Support Vector Machines Classification with a Very Large-scale Taxonomy.” *SIGKDD Explorations Newsletter*, **7** (1), pp. 36–43.

LSHTC3 (Ed.) (2012). *ECML/PKDD-2012 Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*.

McDonald, R., Crammer, K., and Pereira, F. (2005). “Online Large-Margin Training of Dependency Parsers.” In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pp. 91–98.

McDonald, R., Hall, K., and Mann, G. (2010). “Distributed Training Strategies for the Structured Perceptron.” In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 456–464.

Miyao, Y. and Tsujii, J. (2008). “Exact Inference for Multi-label Classification using Sparse Graphical Models.” In *Coling 2008: Companion volume: Posters*, pp. 63–66.

Montejo-Ráez, A. and Ureña-López, L. A. (2006). “Selection Strategies for Multi-label Text Categorization.” In *Advances in Natural Language Processing*, pp. 585–592. Springer.

Punera, K. and Ghosh, J. (2008). “Enhanced Hierarchical Classification via Isotonic Smoothing.” In *Proceedings of the 17th International Conference on World Wide Web (WWW ’08)*, pp. 151–160.

Qiu, X., Gao, W., and Huang, X. (2009). “Hierarchical Multi-Label Text Categorization with Global Margin Maximization.” In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 165–168.

Sasaki, Y. and Weissenbacher, D. (2012). “TTI’S System for the LSHTC3 Challenge.” In *ECML/PKDD-2012 Discovery Challenge Workshop on Large-Scale Hierarchical Text Classification*.

Tai, F. and Lin, H.-T. (2010). “Multi-label Classification with Principal Label Space Transformation.” In *Proceedings of the 2nd International Workshop on Learning from Multi-Label Data*, pp. 45–52.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). “Support Vector Machine Learning for Interdependent and Structured Output Spaces.” In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML ’04)*, pp. 104–113.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). "Mining Multi-label Data." In Maimon, O. and Rokach, L. (Eds.), *Data Mining and Knowledge Discovery Handbook*, pp. 667–685. Springer.

Wang, X.-L., Zhao, H., and Lu, B.-L. (2011). "Enhance Top-down Method with Meta-Classification for Very Large-scale Hierarchical Classification." In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 1089–1097.

略歴

村脇　有吾：2006 年京都大学工学部情報学科卒業．2008 年京都大学大学院情報学研究科修士課程修了．2011 年，同博士後期課程修了．博士（情報学）．同年京都大学学術情報メディアセンター特定助教．2013 年，九州大学大学院システム情報科学研究院助教，現在に至る．計算言語学，自然言語処理の研究に従事．

（2013 年 9 月 30 日　受付）
（2013 年 12 月 8 日　再受付）
（2013 年 12 月 27 日　採録）